ReticularSpaces: Activity-Based Computing Support for Physically Distributed and Collaborative Smart Spaces

Jakob E. Bardram; Sofiane Gueddana; Steven Houben; Søren Nielsen
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen, Rued Langgaardsvej 7, DK-2300 Copenhagen, Denmark
{bardram,sgue,shou,snielsen}@itu.dk

ABSTRACT

Smart spaces research focuses on technology for multiple displays and devices for collocated participants. In most approaches, however, users have to cope with heterogeneous interfaces and information organization, as well as a lack of support for collaboration with mobile and remote users outside the smart space. In this paper, we present *ReticularSpaces*; a multi-display smart space system built on the principles of activity-based computing. The focus of *ReticularSpaces* is to support: (i) unified interaction with applications and documents through *ReticUI*, a novel distributed user interfaces design; (ii) management of the complexity of tasks between users and displays; (iii) mobile users in a local, remote or 'nomadic' settings; and (iv) collaboration among local and remote users. We describe the motivation, design, and architecture of *ReticularSpaces*, and report from a preliminary feasibility study. The study shows that participants found *ReticularSpaces* useful and effective, but at the same time reveals new areas for research on smart environments.

Author Keywords

Distributed User Interfaces, Multiple Display Environments, Smart Spaces, Collaboration, Nomadic Computing

ACM Classification Keywords

H.5.2 INFORMATION INTERFACES AND PRESENTATION: User Interfaces—*User interface management systems*

INTRODUCTION

Technological advances in recent years have led to a myriad of interactive devices and displays, ranging from tiny embedded displays and mobile handhelds to larger and more static tabletops and wall displays. Research into Smart Spaces and Distributed User Interface (DUI) technology seeks to exploit this technology trend by setting up infrastructures that allow users to engage in new forms of seamless digital interaction, which is less tied to specific computing devices, locations or hardware setup, but instead enable tightly integrated group interaction on multiple shared devices and displays. Research on Smart Spaces, such as iLand [16], Gaia [11], Interactive Workspaces / iRos [12] and Impromptu [8] have initiated research into the underlying infrastructures and interaction technologies for such reactive environments. These systems demonstrate how users may utilize new kinds of interaction techniques for using multiple collocated and shared displays and devices.

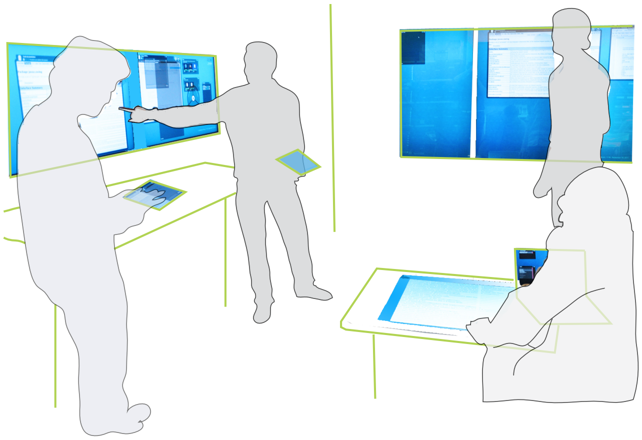


Figure 1. ***ReticularSpaces*** **deployed in a smart room including wall-based, tabletop, desktop, laptop, and tablet displays. Nomadic users can bring devices to the space, which is then automatically added to the smart space setup. Users in other locations can join collaborative sessions from remote displays.**

There is, however, still a set of core research challenges, such as spanning tasks over a large amount of devices or dealing with multiple users, locations and documents [17]. Previous approaches have focused mostly on extending existing personal computing operating systems and user interfaces to work in a collocated device and display sharing setup. This leads to a set of challenges. First, when smart space technology *extends legacy* personal computing operating systems and user interfaces, it implies that users face an interface design and metaphor designed for personal information management at a desk, which in no way is designed to work across multiple computers in the smart space setup. Second, users have *no structured support* for managing large amounts of information, exchange documents or organize flows of tasks between collaborators and devices. None of the existing smart space technologies support users in managing their information according to work tasks, and there is a significant over-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CHI'12, May 5–10, 2012, Austin, Texas, USA.
Copyright 2012 ACM 978-1-4503-1015-4/12/05...$10.00.

head in configuring smart spaces to fit different workflows. Third, the challenge of *reconfiguration* of smart space technology is amplified when mobility is added to the smart space setup, since one of the intrinsic properties of mobile users is that they dynamically join and leave both local and remote collaborative sessions. Most research systems lack support for a dynamic engagement of nomadic users. Fourth, few smart space environments have built-in support for *remote collaboration*, but rely on external tools for awareness, coordination, and communication. This constantly requires an additional effort from users to manually configure collaborative sessions and invite people to join as needed.

In general, the use of legacy user interfaces, and the lack of support for tasks, mobility and collaboration therefore still make this technology disruptive and complex. To address these challenges, we propose the *ReticularSpaces* infrastructure (Figure 1), a new and integrated smart space technology which is based on the principles of *activity-based computing* [3] as the overall design approach. *ReticularSpaces* supports:

1. A *unified interaction paradigm* for handling activities, applications, resources, documents, and services across a distributed displays environment.
2. A *peer-to-peer and event-based architecture* to manage the complex flow of distributed activities and documents between users, displays and devices.
3. Support for *mobile and nomadic* users moving in and out of the smart space environment.
4. Support for *communication, collaboration and awareness* among local and remote users

This paper presents the motivation, design, and technical implementation of the *ReticularSpaces* system, including its distributed user interface and infrastructure architecture. We also report the results of a preliminary feasibility study that was performed in a laboratory setting. We conclude the paper by discussing the lessons learned from implementing and evaluating the *ReticularSpaces* system.

RELATED WORK

Existing smart space research systems can be organized in a taxonomy as illustrated in Figure 2. The four horizontal categories address whether a smart space supports; (i) a user interface (UI) specifically designed for smart space interaction; (ii) document, information, task, and workflows management; (iii) multiple users, in terms of synchronous and asynchronous collaboration; and (iv) mobility, specifically in terms of moving user sessions across devices ('roaming') in multiple locations, and context-aware adaptation as the user moves between multiple locations and hence contexts.

Prior research proposed techniques to transfer digital elements across display boundaries [2, 15]. These approaches were extended to included the relocation of input devices, windows, documents and even applications in a smart space environment. *Aris* for example, provides techniques that support the redirection of windows and input between private devices and public screens [7]. Both the window and input redirection can be used either separately or combined with each screen visualized on an iconic map. Although users can share their content and engage in collaborative work, they are faced with traditional user interfaces which are not designed for smart space setup.

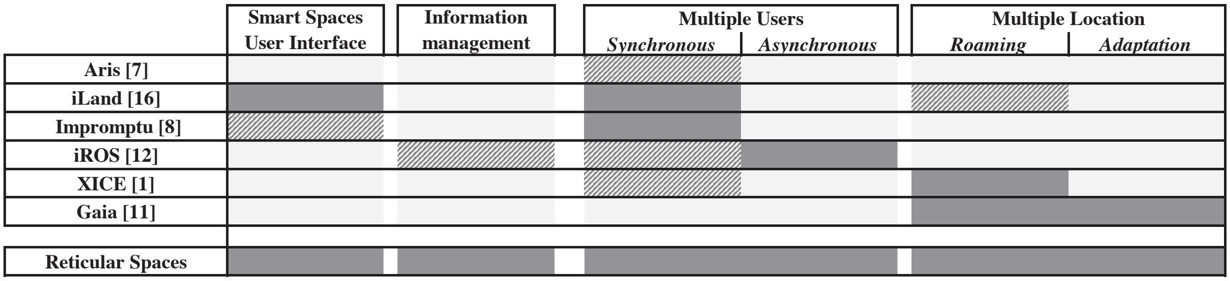
iLand proposed to move beyond legacy interfaces and provided an interface, specially designed for smart spaces, which is integrated in the architectural space and furniture of a smart room [16]. The system allowed users to transfer documents or windows between different screens by moving a physical object between them. Users can also engage in collaborative interactions by replicating their display to computer-enhanced tables and chairs. This allows users to interact simultaneously on multiple displays: users can make remote annotations at the wall display from one of the interactive chairs or manipulate an artifact at the table. The iLand system provided a *unified* interface enabling collocated and synchronous group collaboration.

Impromptu was designed specifically to support collocated collaborative interaction in smart spaces. Smart space users can exchange application windows by replicating them, e.g. for problem solving, or by sharing them on public displays, e.g. for discussion or reflection [8]. To improve collaborative smart space experiences, Impromptu integrates special collaboratives tools, such as tele-pointers, screen sharing and instant messaging, into the smart room technology. However, there is currently no support for asynchronous data access or remote and nomadic collaboration.

Other research approaches extended smart room technology to include dynamic user membership and mobile devices. *IRos* is a suite of systems that can be used to create applications for multiple devices with the ability to integrate portable devices in smart space interaction [12]. It supports the redirection of input with *PointRight* [13], replication of content with *Multibrowse* [12], and asynchronous exchange of documents with *DataHeap*. In iRos, information can thus be accessed across multiple displays and by mobile users that are dynamically added or removed as they join or leave the smart space.

Focusing specifically on nomadic computing, *XICE* supports dynamic engagement. It allows users to easily extend their mobile device input and output by *annexing* smart space wall and table displays [1]. Hence, XICE support basic device roaming. In a multi-user setup, XICE allow for several people to annex the same screen, but not to merge their displays and inputs into one shared collaborative surface. *Gaia* builds on the same annexation principle, but treats each user as a physical object that triggers the addition of information into a smart space [11]. Data are directly linked to the user, thereby becoming available whenever they enter a new active space. The system also supports multiple displays by rebinding whole applications to any display or device. User context is used to organize data and make information that is relevant to the current context easily accessible.

Outside smart space research, Activity-Based Computing (ABC) supports complex information and task management, and has been successfully applied to several areas, including desktop computing [5, 18] and pervasive computing in hospitals [3,



	Smart Spaces User Interface	Information management	Multiple Users		Multiple Location	
			Synchronous	*Asynchronous*	*Roaming*	*Adaptation*
Aris [7]			▨			
iLand [16]	■		■		▨	
Impromptu [8]	▨		■			
iROS [12]		▨	▨	■		
XICE [1]			▨		■	
Gaia [11]					■	■
Reticular Spaces	■	■	■	■	■	■

Figure 2. Research systems support for smart spaces challenges (limited support is dashed).

4]. ABC proposes to aggregate relevant resources into activities and provides a set of principles that describe how activities can be used [3]. Recently, Clinical Surfaces [4] used the ABC approach to allow clinicians to manage, access, and move patient data across a distributed multi-display environment covering a whole hospital. Based on the ABC principles of 'activity-centric resource aggregation' and 'activity suspend/ resume', Clinical Surfaces aggregates medical information in working contexts and dynamically presents information and tasks relevant for a user context.

Compared to previous research on smart space technology, *ReticularSpaces* presents an integrated approach to a unified UI design, information and task management, collaboration, and nomadic computing that builds on the activity-based computing approach and principles. Compared to previous work on ABC, *ReticularSpaces* extends the research on distributed multi-display environments for hospitals to a generic approach for general-purpose smart space technology.

RETICULAR SPACES – APPROACH AND CONCEPTS

The goal of *ReticularSpaces* is to provide a novel smart space infrastructure and UI that unifies task and information management with support for mobility and collaboration in a distributed user interface environment. The principles of activity-based computing (ABC) [3, 6] works as the unifying concepts and metaphors. Thus, *ReticularSpaces* extends the use of activity-based computing to a multiple display environment. This section describes the overall approach and concepts of *ReticularSpaces* with respect to providing a uniform information and task management, remote and local collaboration and mobility support. The technical description of the UI software technology and underlying infrastructure of *ReticularSpaces* is presented in the following sections.

Unified Information and Task Management

Unified information management in *ReticularSpaces* is based on the ABC principle of 'Activity-centric Resource Aggregation'. This principle states that all documents, resources, services, etc. that are relevant for a human activity, should be organized into a corresponding 'Computational Activity', or just 'Activity'. Unified task management is based on the ABC principle of 'Activity Resume/Suspend', which states that activities have multiple participants, which each can resume and suspend an activity and thereby work on it.

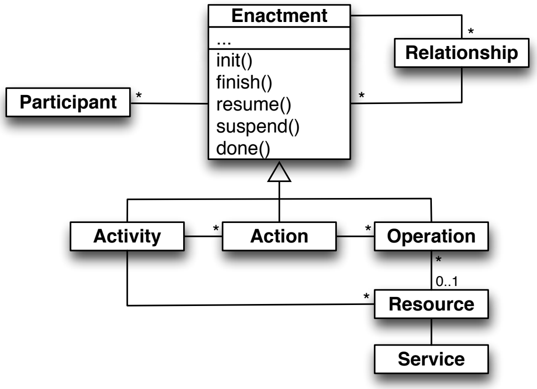


Figure 3. The ABC Ontology used in *ReticularSpaces*.

ReticularSpaces implements these ABC principles into an ontology illustrated in Figure 3. Each *activity* is composed of a set of *actions*, each again holding a set of *operations*. Each operation points to a *resource*, such as a document, a picture, html page, etc. Resources can also be external *services*, such as a device, like a printer, which can be accessed through an operation. Each activity has a list of *participants*, and only participants can access (resume/suspend) the activity, and its actions and operations. *Relationships* allows users to organize activities, actions, and operations in different workflow structures. Such structures could be simple association links showing which activities are related, as well as more complex workflow constraints specifying which activities has to be done before another activity can be resumed. Because of the overlapping properties between activities, actions and operations, they are abstracted into an *enactment*.

Shared Activities and Collaboration Tools

Since an activity (and its actions and operations) can have several participants, they are by default shared and collaborative. This is the ABC principle of 'Activity Sharing'.

Asynchronous activity sharing happens when participants take turn in resuming and working on an activity. Since *ReticularSpaces* maintains state information about the work done in activities, one participant can take over working where the previous participant left the work. By using an *activity log*, users can get a log over past events including log entries entered by other participants. Hence the activity log also works as a medium for leaving activity specific messages to each other.

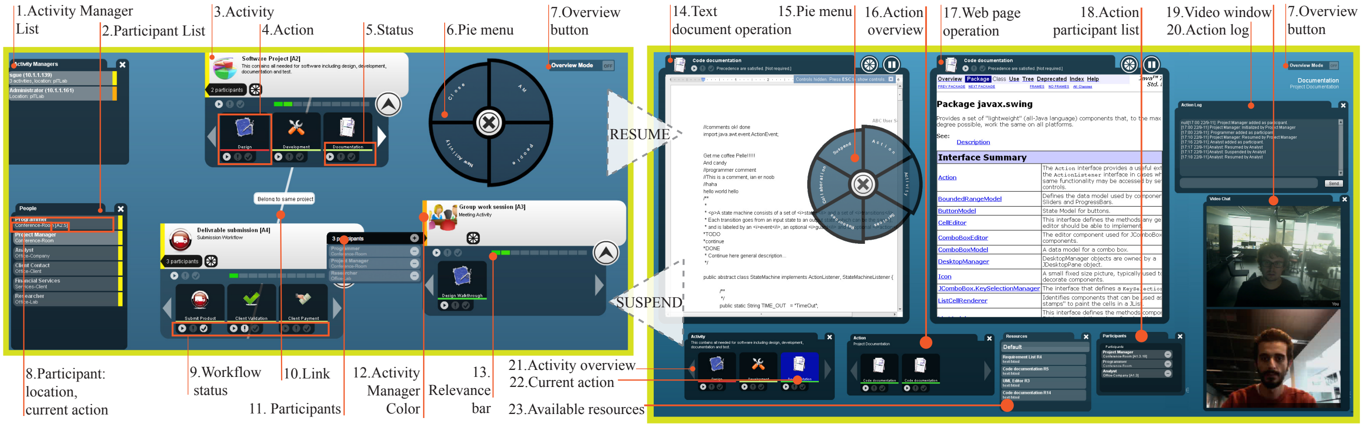


Figure 4. The *ReticUI* user interface. The *Activity View* (on the left) shows a list of activity managers available, a list of users in this smart space room, and the set of activities that are relevant in this context. Each activity (the while box) can be expanded to show its list of actions and participants. Workflow relationships between activities are shown as lines with a text label. The *Action View* (on the right) is displayed when a user resumes an action. This view shows the action's operations and the resource each operation links to, such as a text document or a web page. The action view can show various overview panels as shown at the bottom of the view. From left to right these are overviews of: all actions in the overall activity; operations in this action; available resources; and the participants. On the right side the collaboration windows are shown, including (from the top); the action log and the remote video feeds. Users can switch between the two views by suspending and resuming an action. All user interaction works by multi-touch gestures and pie menus.

If two or more participants resume the same activity simultaneously from multiple displays – either collocated in the smart space or remotely – they engage in *synchronous activity sharing*. Due to the shared state management this entails that the view on the displays are synchronized (as in a desktop sharing setup) and appropriate communication channel like text, voice, and video are established if needed. Since the activity log is updated in real time, it works as an instant messaging system in this case. The activity as such hence work to mediate a collaborative space and since an activity knows it list of participants, automatic session management is achieved.

Mobility and Context-Awareness

ReticularSpaces supports nomadic work and mobility by implementing the ABC principle of 'Activity Roaming'. This principle states that a participant should be able to suspend an activity on a device in one location and then later resume it on a display in another location. The activity is resumed in the same state (and hence looks the same) as when suspended, since the activity maintain state information.

ReticularSpaces has build-in support for location tracking using active badges. Similar to ClinicalSurfaces [4], this information is used in context-aware adaptation of the UI to show the most relevant activities for the set of users in a specific location. Hence, a wall display in a smart space will mainly show the activities that are relevant for the people in the room, which typically are the activities they participate in. The 'relevance' mechanism is rather flexible and can be extended to also include the collocation of e.g. objects, devices, and non-users. The list of participants of an activity will also show their location information, which helps provide an awareness of the whereabouts of participants in an activity.

Location of users is also used to provide context-aware voice and video connectivity in a synchronous activity sharing session. If the participants are in different places, then an audio and video link is automatically established – something that is not needed if they are collocated. If they move to the same location, the communication link is removed.

As we shall described below, the architecture of *ReticularSpaces* is a peer-to-peer architecture where each device can store activity information in a local *activity manager*. In order to share local activity information, *ReticularSpaces* allow displays to mount activity managers as they are discovered. Hence, if a user with a laptop enters a smart space, all devices will discover the activity manager of each other. The user can then, for example, go to the public display and mount his own activity manager, thereby getting access to activities and data on his own laptop. Access to an activity manager can be secure and password protected. When leaving the smart space again, the user's activity manager is automatically unmounted.

RETICUI: AN ACTIVITY-BASED UI FOR SMART SPACES

Designed to unify the user experience in a smart spaces, *ReticularSpaces* provides the *ReticUI* user interface (UI) as shown in Figure 4. *ReticUI* consists of two views; the Activity View and the Action View. Both views are zoomable and panable, and are thus designed to be used on any display size, even very large wall-based displays. Contextual pie menus are used consistently to allow for users to access them across a large wall-based display. The displays do not, however, support rotation, and when used on a tabletop display, the user should be situated in 'front of' the table.

Activity View

The activity view is the default view that is shown on a display. This view initially shows a list of people and activity manager in the same location as the display; the default activity manager is always the local one. Users can access a display and mount an activity manager by clicking its icon and entering a PIN code. The activity view will then display activities, from this activity manager, that are relevant for the current context. The current implementation of the relevance algorithm finds the activities that people in the room participate in, but the architecture is open for implementing more advanced relevance algorithms. The degree of relevancy is show in the green relevance bar. In Figure 4, three activities are shown which are related to a software company, and Figure 5 shows an activity in detail. Since several activity managers can be mounted on a display, the left border of an activity is marked by a color reflecting which activity manger it belongs to. By clicking the black arrow, the activity expands to show its list of actions beneath it, and by clicking the 'participants' label, the list of participants are shown to the left of the activity. By using the small '+' and '−' buttons, a participant can add or remove participants to the activity.

Workflow relationships between activities are shown as lines with a text label. Workflow status for both the overall activity and for each of the actions are revealed by the three small icons beneath each activity or action; the 'play' icon indicate if this activity/action can be resumed; the '!' icon indicates if this activity/action is required (i.e., has to be done), and the 'checkmark' icon indicates whether the activity/action has been done. The description of the activity-based workflow features of *ReticularSpaces* is, however, outside the scope of this paper.

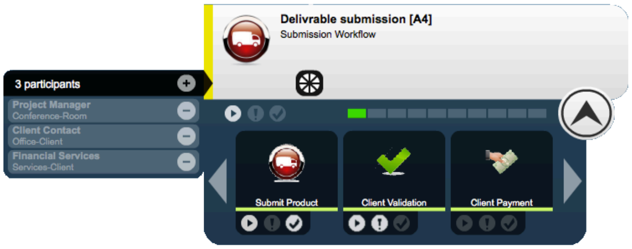


Figure 5. The activity as shown in the activity view. Each activity has an icon, a name, an ID, a description, a set of participants, a list of actions, and a workflow status. Each action has an icon, a name, and workflow status. The color bar beneath each action indicates its status (resumed/suspended).

Action View

When a user clicks an action icon, *ReticUI* shows the action view (Figure 4, right). The primary purpose of this view is to show the action's operations and the resources they link to. Since work is done at the level of operations, the action view is where the user is working with the resources in the action. Resources can be text documents, images, files, or web pages, as shown in Figure 4. A resource can also be an external service, like a printer or a video camera, in which case the operation will show a user interface suited for this service. Operations can be created, copied, and deleted, and can be marked as done, which is then reflected in the overall workflow status. Like activities and actions, operations can be resumed and suspended. Suspended operations in an action cannot be accessed. The action view has a set of auxiliary panels that provide meta information to the user. At the bottom, from the left in Figure 4 is shown: the activity panel showing an overview of the activity that this action is part of; the action panel showing an overview of this action and its operations; a list of available resource in this activity (resources are managed on the level of the activity); and a list of participants involved in this activity, including their current work activity, status, and location. On the right-hand side of the activity view are two collaboration panels: the action log showing both a history of events happened in this action and text entries from users; and the video window showing video feeds from participants currently having this action resumed – in this case two other users also participate. As mentioned above, this video link is context-aware and is only established if the users are not collocated.

As the action view is zoomable and panable, the view consists of a large canvas on which numerous operations are displayed. Zooming and panning is done with standard pinch-to-zoom and drag-to-pan gestures. The overview button is used on single-touch devices to zoom out and show all operations.

By using state synchronization between devices that are part of a synchronous activity sharing session, the action view is kept synchronized across the participating displays. This applies for the placement, size and content of the operation windows. This entails that a user working on a large, high resolution display may move operations into places where participants using smaller devices cannot see them. However, by using the zoomable canvas, users are always able to zoom out, get an overview of all operations and zoom in and focus on an area appropriate for them. Note that there is no 'zoom mode' in *ReticUI*; all functionality, buttons, icons, and pie menus work regardless of zoom level. Since the auxiliary panels work as personal toolboxes for a user, they are neither zoomable or synchronized during an activity sharing session.

ARCHITECTURE

The software architecture of the *ReticularSpaces* system is illustrated in Figure 6. The system consists of two main components; the *ReticUI* user interface component and an underlying infrastructure consisting of a set of Activity Managers. In addition to these two components, the infrastructure is able to connect to external services, like location tracking systems, RFID readers, and other domain-specific services.

The *ReticularSpaces* infrastructure is a peer-to-peer, event-based distributed system. Peer-to-peer means that a copy of the infrastructure – the Activity Manager – is deployed on all devices participating in the system setup. Communication, data sharing, and event propagation takes place directly between the devices participating in the smart space setup. *ReticularSpaces* hence do not use a centralized server, but one device can act as a 'super-peer' and de-facto work as a server, if needed. Event-based means that the infrastructure supports a Model-View-Controller (MVC) architecture [10] based on subscribe-notify type of interaction, rather than the more traditional request-response model. Event-based architectures are used in user-interface technology when updating user interface components based on events from e.g. input devices. As *ReticularSpaces* is a distributed user-interface system, it similarly rely on an extended event-based style of programming now running distributed across a network. Hence, the

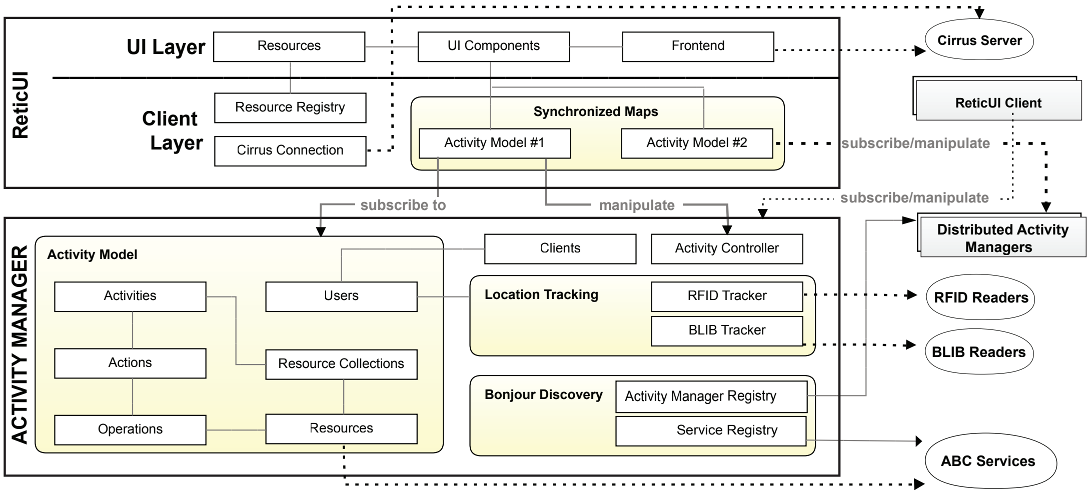


Figure 6. The software architecture of *ReticularSpaces* consisting of the *ReticUI* UI component (shown in Figure 4) and the Activity Manager infrastructure. The *ReticUI* connects to a local or remote Activity Manager, and Activity Managers can connect to each other in a distributed peer-to-peer fashion. Activity Managers and other application-specific services are discovered on the network using mDNS/Bonjour service discovery.

ReticularSpaces architecture essentially implements a distributed Model-View-Controller (dMVC) architecture [9, 14].

The main components of the Activity Manager are the Activity Controller and the Activity Model. The activity model organizes data into the ABC ontology shown in Figure 3. The activity controller holds all data on users, devices, activities, resources, etc. The activity model and the activity manager are based on the Aexo framework, which implements a distributed model-view-controller architecture [9]. A *ReticUI* client can register to one or more activity managers and by using the distributed event-based mechanisms in Aexo, the client maintains a synchronized copy of the activity model from each manager. Updates to the local activity model then again updates the *ReticUI* accordingly, using a local observer pattern. The activity controller provides an interface for external clients, including the *ReticUI*, to control the activity manager and the activity model. It provides mechanisms for managing users, clients, services, and other activity managers.

Each activity manager announces itself on the network using Bonjour (mDNS) and activity managers can hence automatically discover each other. Discovered activity managers are registers in the Activity Manager Registry and listed in the Activity Manager List in the *ReticUI* (shown on the left hand side of Figure 4). From this list the user can then select which activity manager he wants to 'mount' and thereby connect to. These mechanisms of activity manager discovery and mounting is used to set up and maintain the peer-to-peer topology of activity managers in a smart space.

Similarly, external services can be discovered and registered in an activity manager. For example, location tracking of users is done by having location tracking services discovered and registered as a service to an activity manager. These services then update the data model with location information about users and devices.

A domain-specific service can be associated with a resource. When an action is resumed, automatic service discovery of domain-specific services linked to the resources of the action's operations are initiated. When the operation is resumed, the service is asked to also resume working. A service can be a virtual service such as a web service, or a physical device such as a scanner or a printer. The *ReticUI* client is just handed an URI pointing to the service, and uses this URI to fetch and show any data coming from that service. This flexible mechanism separates domain-specific data from the *ReticularSpaces* infrastructure itself and routes data to/from the *ReticUI* directly to/from independent services. This mechanism account for the dynamic addition, removal and network relocation of devices.

The *ReticUI* client holds a synchronized replication of the activity data model for each of the activity managers to which it has registered. On top of this, the *ReticUI* implements the different UI components shown in Figures 4. *ReticUI* is designed to work across all types of devices and displays in a smart space. As such, the goal of *ReticUI* is to provide a homogeneous user experience in manipulating and displaying data elements like activities, actions, operations, resources, and services.

For each type of domain-specific resource that *ReticularSpaces* handles, *ReticUI* provides a corresponding resource view component that is able to render the resource inside the operation windows in the action view. For example, in order to show HTML resources, an HTML resource render (i.e., a web browser) is needed. Domain-specific resource components are loaded on runtime and registered in the Resource Registry of *ReticUI*. When an action, its operations and associated resources are resumed, the corresponding UI resource components are looked up in the resource registry. This is based on the type of a resource. And since a resource just points to a service, the UI resource component knows how to connect to a service, and render and manipulate data from this service. For example, the service shown in Figure 4 shows a action view with a text resource, which is pointing to a shared text document residing on a server. Note that sharing of resources is not part of the architecture, and resources will normally reside on servers external to *ReticularSpaces* .

As discussed above, support for collaboration is a core part of *ReticularSpaces*. Synchronized action view and real-time update of the activity log is supported by a distributed state synchronization mechanism. This is implemented via the distributed MVC architecture of Aexo, which maintains synchronization of the distributed activity models. This implies that any update to an activity model on one client will be synchronized on all clients connected to the same activity manager, and updates to an activity model will in turn cause an update of the user interface. Hence, support for synchronized views while collaborating is a built-in feature of the distributed MVC architecture. Video conferencing is based on an external video server, acting like any other external service in the system; when resuming an action with a 'video resource' the video UI component connects to the video server and starts streaming video content to and from it.

The *ReticularSpaces* architecture is implemented using different technologies. The Activity Manager is implemented in Java on top of the Aexo distributed event system, using a RESTful[1] protocol for event distribution and XML for serializing the activity model. The *ReticUI* is implemented in Adobe Flex AIR[2], and the video server is using the Adobe RTMFP Cirrus server[3]. Location tracking is done by the BLIP Bluetooth location tracking system[4], and the RFID tagging is done using the WaveTrend activity RFID technology[5]. Resources are handled using uniform resource identifies (URI) and MIME types.

FEASIBILITY STUDY

In order to evaluate the *feasibility* of the *ReticularSpaces* approach and verify the *stability* of the system before moving into larger scale field experiments, the system was deployed as a smart space environment in a research lab as shown in Figure 1. The goal of this study was to expose users to the design of this new smart space technology and assess its perceived usefulness. To focus on the features specific to our complex system, we conducted a feasibility study rather than comparing it with previously implemented systems. Essentially, we were interested in evaluating the following question: "Does *ReticularSpaces* effectively support interaction with multiple displays and devices, involving multiple users distributed across multiple locations?"

Participants and Setup

12 participants (mean age 27) were recruited to participate in the experiment. The setup consisted of two large interactive wall displays, a multitouch tabletop display, two tablet computers, and a notebook. The smart space itself containing the large wall displays and the tabletop display functioned as a meeting room, while the mobile devices (i.e., the laptop and tablets) moved in and out of the smart space or were used in the hallway, another meeting room, and a remote office.

Method

As both the system as well as the concept of a smart space environment were completely new to most users, we used a scenario-based approach to guide participants in interacting with *ReticularSpaces*. After a short 15 min. introduction to the basic functionality and design rationale of the system, participants completed 5 scenarios in groups of 2 using the think-aloud method. Scenarios were randomized per group of 4 participants using latin square. The first scenario was excluded from randomization, as it included basic notions required to complete other scenarios. Completing all scenarios took approximatively one hour for each group. During the completion, participants were observed by one researcher while being guided through the scenario by another researcher. After each scenario, participants filled in a 5-point Likert scale questionnaire, and a semi-structured interview was conducted with them after completing all scenarios. During the interview, participants elaborated on their questionnaire results and answered open questions related to the user experience.

Scenarios

Five scenarios were designed to represent real situations that occur in a typical software development company. The first two scenarios were focused on information management: (1) create a presentation at office before presenting it at the conference room and sharing it with the team; and (2) collaborate on completing a project delivery by taking turns in performing actions, such as "submit deliverable' and "validate submission'. In the remaining three scenarios, participants were introduced to different collaborative setups: (3) working asynchronously on the code of a software project while communicating code changes; (4) starting a group meeting remotely on a mobile device before arriving into the conference room thereby changing from remote to local settings; and (5) moving to a conference room and move work from a tablet computer to the wall display, while discussing shared artifacts with a remote colleague. Scenario 4 is shown in Figure 7 as an example.

Focus of the experiment

Questionnaires and interviews focused on the following four areas:

Activity-Based computing. The usefulness of the basic concepts of activity-based computing such as a the main components of an activity (activity, action and operation), the suspending and resuming of activities, the management (activity creation, templates and relations), and the activity workflow features.

Distributed Interaction. The usefulness of activity roaming across different locations, different devices, and access to distributed resources such as documents.

Asynchronous Collaboration. The usefulness of the notion of multiple participants, the action log, and the sharing of activities.

[1] `http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm`

[2] `http://www.adobe.com/products/air/`

[3] `http://labs.adobe.com/technologies/cirrus/`

[4] `http://www.blipsystems.com/`

[5] `http://www.wavetrend.net/`

Scenario 4 – *Late for a meeting, the project manager (PM) resumes the meeting activity on his tablet PC, while walking down the hallway. The activity view of* ReticUI *shows the shared documents and automatically establishes a video link with his colleagues in the meeting room, allowing him to announce that he is on his way. Arriving in the meeting room, the video connection is automatically stopped. Meanwhile, a remote colleague (RC) sitting in his office notices that his colleagues are currently working on their common project in the meeting room. She decides to join them. When she resumes the action, a video link is established connecting her with the meeting room, and her colleagues fill her in on their discussion. RC starts adding text to one of the shared documents, and her colleagues in the meeting room see the text appear in their screens. Then, while discussing two version of a diagram, PM place them along each other from his tablet, which update documents layout on all displays. Afterward each team member start to work on different parts of a document thereby being highly focused on their individual tasks. To ask a question without interrupting others, RC uses the action log and type it in a chat message. PM notices the question on the wall display and answers it with a chat message from his tablet. After the meeting, back to his office, PM resumes the meeting activity and all resources are restored as suspended in the smart room.*

Figure 7. Scenario 4 focusing on remote group meeting.

- *Synchronous Collaboration*. The usefulness of real-time collaborative instant messaging, video conferencing and synchronous activity sharing, as well as the usefulness of the list of participants while collaborating.

Results

Figure 8 provides an overview of the quantitative results of the questionnaire. The results from the interviews are discussed in the 'Discussion' section.

Activity-Based Information Management

In general, the activity-based concepts were perceived as very useful ($Q3$: μ= 4,2; $\tilde{x}$= 4; σ= 0.72). Restrictions of workflows on action level (with the action view) was seen as a good approach to deal with distributed work that requires some order ($Q4$: μ= 4,25; $\tilde{x}$= 4; σ= 0.63). Participants reported that sharing documents easiness was average ($Q1$: μ= 3,6; $\tilde{x}$= 4; σ= 1,165). Work management was also rated with an average score ($Q2$: μ= 3,5; $\tilde{x}$= 3; σ= 0.79). We anticipated that this average score was caused by a learning curve and this was confirmed during the interviews.

Distributed Interaction

The fact that participants were able to access activities and their resources on several devices simultaneously was considered very valuable ($Q7$: μ= 4,5; $\tilde{x}$= 4.5; σ= 0.52). The location roaming ($Q5$: μ= 4,6; $\tilde{x}$= 5; σ= 0.49) and device roaming ($Q6$: μ= 4,4; $\tilde{x}$= 5; σ= 0.79) were also found highly useful.

Asynchronous Collaboration

In general, participants found the action log very valuable both for leaving messages related to the action log ($Q8$: μ= 4; $\tilde{x}$= 4; σ= 0.58) as well for automatic logging ($Q9$: μ= 4,1; $\tilde{x}$= 4; σ= 0.2).

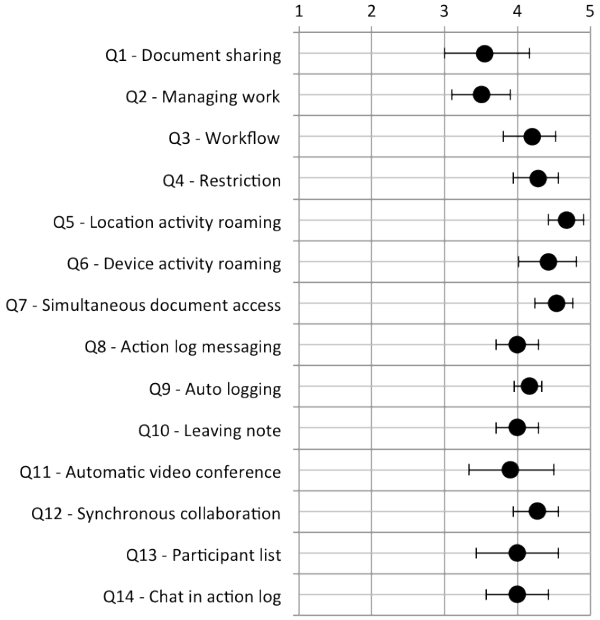


Figure 8. The results of the questionnaire. Scores are on a 5-point Likert scale from 1 (not useful) to 5 (very useful).

Synchronous Collaboration

Overall, participants found the system very useful for synchronous collaboration ($Q12$: μ= 4,3; $\tilde{x}$= 4; σ= 0.62). The display of the list of participants was also considered useful ($Q13$: μ= 4; $\tilde{x}$= 4; σ= 1,13) as it could be a starting point for collaboration. The use of the action log as an additional real-time communication channel was also appreciated as useful ($Q14$: μ= 4; $\tilde{x}$= 4; σ= 0.85). Finally, most participants agreed that automatic video conferencing is a good way to support collaboration with a remote person ($Q11$: μ= 3,9; $\tilde{x}$= 4; σ= 1,13).

DISCUSSION

The purpose of *ReticularSpaces* is to unite information management, co-located and remote collaboration and mobility into a new activity-based smart space environment. In this section, we discuss these features based on the results of the study questionnaires and interviews.

Activity-based Information and Task Management

One of the key challenge that we addressed in *ReticularSpaces* is unifying task and information management in smart spaces into a new activity-based UI metaphor, *ReticUI* . By introducing a homogeneous interface that allows for a seamless transition between synchronous and asynchronous work, *ReticularSpaces* can support dynamic and multi-device smart space environments. Information can easily be accessed, moved and shared between smart space users but also between devices. In the interviews, several participant expressed their appreciation for this unified interface and argued that the integration between individual and collaborative work, as well as between collocated and remote participants felt "natural".

During interviews, the activity-based organization of work in our system was considered especially valuable for managing large amount of information. One participant referred to this type of interface as a "project-based interaction with information and users". However, several users were confused that they did not need to save documents, as everything is automatically synchronized, and explained that this may be due to different habits. Clearly the new paradigm implies a learning curve, but we observed that participants became familiar with the approach before the end of evaluation sessions. In the interviews, one participant explained that he found this organizational approach of *ReticularSpaces* much more natural then traditional computing systems.

Our activity-based approach also introduces workflow relationships and constraints that helps build and manage complex flows of tasks. The study showed that this approach can greatly aid different roles in complex collaborative set-ups but that the current implementation was too limited. Some participants expressed concerns regarding the scalability of this feature if it were to be used in large and complex organizations. A remaining issue for future work in the UI design is thus the interactive visualization and construction of such activity-based workflows. *ReticUI* represents workflow steps simply as successive action visualized with a status bar. Participants pointed out that even though workflow status is shown in the UI, it is hard to grasp the overall logic and organization of a workflow, specially in large or complex projects. But in general the results of the study provide evidence that activity-based computing is a valuable approach for a unifying design concept for smart space technology.

Activity-mediated Collaboration

Another core goal of *ReticularSpaces* is to introduce the smart space environment as a technology for collaborative work among both co-located and remote users. Beside support for an interface and information management model that supports multi-device co-located and remote collaboration, *ReticularSpaces* also provides a set of tools to directly support collaboration and awareness.

For example, the list of users provides an overview of all users, and shows what activity (and action) they are currently engaged in as well as their current location. Several of the study participants argued that this is an excellent mechanism for starting a new collaboration session, as location can be an important factor for collaboration. For example, when an employee working from home sees that several other colleagues are having a discussion in the meeting room, this might trigger him to quit his ongoing activity and remotely join the discussion. Additionally, the list of an activity's participants is very helpful to understand who is currently engaged in the activity.

The use of the action log for posting notes in asynchronous communication was also found useful. In the interviews, participants suggested several applications for this asynchronous communication channel, such as task delegation, communicating changes in the shared action, or requesting updates to a document. For synchronous communication, the seamless transformation from an individual activity to real-time synchronous activity sharing was perceived as very useful. Furthermore, the use of automatic video conferencing as an additional communication channel for remote collaboration was reported to be a very strong feature, because it eliminates the need to start separate communication applications and establish the relevant collaboration sessions with all participating users. As one user put it:

> — *"The integrated collaboration is really useful. Even in cases when you are sick at home, you can still fully participate in meetings without the need for you and your colleagues to setup many tools."*

ReticularSpaces uses location information to maintain an audio-video communication channel only when users are engaged in a remote collaborative setting. The link is thus only established when remote participant joins the activity and is removed when she leaves. During the interviews, some participants argued that automatic video is a good approach because video adds a layer of trust to the collaboration. However, some participants also mentioned that there might be privacy issues when using an automatic video connection.

The study also showed that instant messaging in the action log can work as a real-time communication back channel used for aside or less intrusive communication that supplements the video communication. By attaching communication channels to shared structure that also holds users' documents and tasks, these features of *ReticularSpaces* hence introduce an efficient approach for collocated and remote collaboration.

Mobility and Context Awareness

One of the critical differences between *ReticularSpaces* and most of previous work is the integration of *mobility* into the smart room environment. A user can simply walk in and out of a smart room while carrying his work with him. In *ReticularSpaces*, mobility is implemented through activity roaming and activity suspend/resume. This allows a user to suspend, move, and resume an activity between locations and/or devices, or even allow for simultaneously work on several devices. This feature was really appreciated by the participants of the user study. Most of our participants mentioned these features as the best of the system. During the interviews, one participant suggested that *ReticularSpaces* might be able to solve a common challenge in meeting rooms:

> — *"You can simply prepare your work at your office, walk into the conference room, wirelessly mount your device to the room, present your work and afterwards share the presentation with the room display with people interested in having it. This kind of interaction and information sharing is currently a big problem at our company."*

By using location tracking, only activities that are relevant to the active participants are presented on the displays in the smart room. This mechanism allows for a scalable activity visualisation as it implicitly deals with cluttering but also provides integrated and fluid support for mobile users as they walk in and out of smart rooms.

CONCLUSION

ReticularSpaces is a multi-display smart space system that uses the principles of activity-based computing to offer: a novel unified UI named *ReticUI* for smart space technology; support for management of complex tasks; an extension of smart space interaction to mobile users; and collocated and remote activity-mediated collaboration.

In this paper, we presented the motivation, design, and architecture of *ReticularSpaces*, and reported from a preliminary feasibility study. The study showed that participants found the system and approach very useful, while at the same time uncovered a set of challenges to be addressed in future research on smart space technology. Future work will refine *ReticularSpaces* design and prepare it for a large scale field study.

REFERENCES

1. R. Arthur and D. R. Olsen, Jr. Xice windowing toolkit: Seamless display annexation. *ACM Transactions on Computer-Human Interaction*, 18:14:1–14:46, August 2011.
2. T. Bader, A. Heck, and J. Beyerer. Lift-and-drop: crossing boundaries in a multi-display environment by airlift. In *Proc. of AVI 2010*, pages 139–146, New York, NY, USA, 2010. ACM.
3. J. E. Bardram. Activity-based computing for medical work in hospitals. *ACM Transactions on Computer-Human Interaction*, 16(2):1–36, 2009.
4. J. E. Bardram, J. Bunde-Pedersen, A. Doryab, and S. Sørensen. Clinical surfaces — activity-based computing for distributed multi-display environments in hospitals. In *Proc. of INTERACT 2009*, pages 704–717, Berlin, Heidelberg, 2009. Springer-Verlag.
5. J. E. Bardram, J. Bunde-pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proc. of SIGCHI*, pages 211–220. Press, 2006.
6. J. E. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI 2006*, pages 211–220, New York, NY, USA, 2006. ACM Press.
7. J. T. Biehl and B. P. Bailey. Aris: an interface for application relocation in an interactive space. In *Proc. of GI 2004*, pages 107–116, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
8. J. T. Biehl, W. T. Baker, B. P. Bailey, D. S. Tan, K. M. Inkpen, and M. Czerwinski. Impromptu: a new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proc. of CHI 2008*, pages 939–948, New York, NY, USA, 2008. ACM.
9. J. Bunde-Pedersen. *Distributed Interaction for Activity-Based Computing*. PhD thesis, Computer Science Department, University of Aarhus, Denmark, 2009.
10. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
11. C. K. Hess, M. Román, and R. H. Campbell. Building applications for ubiquitous computing environments. In *Proc. of Pervasive 2002*, pages 16–29, London, UK, 2002. Springer-Verlag.
12. B. Johanson, A. Fox, and T. Winograd. The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing*, 1:67–74, April 2002.
13. B. Johanson, G. Hutchins, T. Winograd, and M. Stone. Pointright: experience with flexible input redirection in interactive workspaces. In *Proc. of UIST 2002*, pages 227–234, New York, NY, USA, 2002. ACM.
14. N. Marquardt and S. Greenberg. Distributed physical interfaces with shared phidgets. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, TEI '07, pages 13–20, New York, NY, USA, 2007. ACM.
15. J. Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proc. of UIST 1997*, pages 31–39, New York, NY, USA, 1997. ACM.
16. N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz. i-land: an interactive landscape for creativity and innovation. In *Proceed. of CHI 1999*, pages 120–127, New York, NY, USA, 1999. ACM.
17. L. Terrenghi, R. May, P. Baudisch, W. MacKay, F. Paternò, J. Thomas, and M. Billinghurst. Information visualization and interaction techniques for collaboration across multiple displays. In *Extended Proc. of CHI 2006*, CHI '06, pages 1643–1646, New York, NY, USA, 2006. ACM.
18. S. Voida, E. D. Mynatt, and W. K. Edwards. Re-framing the desktop interface around the activities of knowledge work. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 211–220, New York, NY, USA, 2008. ACM.